Vol. IV, No. 2, mayo - agosto, 2013, pp. 37 - 42

Propuesta de algoritmo de clasificación genética

José Leandro González
Correo electrónico:jlgonzalez@uci.cu

Artículo original

Omar Mar Cornelio
Correo electrónico:omarmar@uci.cu.cu
Universidad de las Ciencias Informáticas (UCI), La Habana, Cuba

Resumen
La extracción de conocimiento representa la base de la toma de decisiones en diversos procesos; siendo la minería de datos un área del conocimiento fundamental para garantizar dicho objetivo. Entre los procesos más utilizados, en la actualidad se encuentran la inteligencia artificial y el análisis estadístico implementado mediante técnicas de asociación, agrupamiento, clasificación que son desarrolladas con la aplicación de métodos estocásticos, heurística, redes neuronales. Sin embargo, ante problemas de análisis biológicos dan soluciones temporales representando bajo grado de generalización. La presente investigación describe una solución a la problemática planteada incidiendo sobre la técnica de clasificación a partir de la creación de un algoritmo de programación genética para la generación de reglas de producción, para lo cual se proponen cuatro fases en su implementación donde se transita desde la iniciación, pasando por la evolución de lo individuos, evaluación de las poblaciones hasta la culminación. Se valida además la eficiencia del algoritmo y se hace un análisis comparativo sobre las soluciones LogenPro y Pgirla demostrándose la eficiencia de la propuesta presentada.

Palabras claves: algoritmo, clasificación, programación genética, reglas

Recibido: 25 de marzo del 2013 Aprobado: 4 de abril del 2013

INTRODUCCIÓN

La minería de datos es actualmente un tema muy difundido, que engloba una serie de procesos y técnicas muy novedosas basadas en la inteligencia artificial y el análisis estadístico, encaminados a la extracción de *conocimiento* procesable, nuevo y útil, implícito en grandes almacenes de datos y/o bases de datos. Para encontrar dicho conocimiento implícito en la información, esta rama se apoya en la realización de diferentes tareas, en dependencia del problema que se plantee, tales como: asociación, agrupamiento (*clustering* en ingles), clasificación, entre otras, siendo esta última una de las más usadas. Como métodos de solución a estas tareas se encuentran los algoritmos estadísticos, redes neuronales, algoritmos evolutivos o algoritmos de búsqueda basados en reglas probabilísticas, entre otros, y estos últimos presentan variantes como los algoritmos genéticos y la programación genética. Dada la efectividad de las tareas antes mencionadas, se sugiere su aplicación en herramientas que apoyen las investigaciones realizadas en diferentes áreas científicas, ya que permiten procesar grandes bases de datos y son capaces de manejar indisolublemente datos de altas complejidades. Los paradigmas evolutivos actuales están inspirados en la teoría de la evolución de Darwin e intentan simular en lo posible los procesos que ocurren en la naturaleza, basando la resolución de problemas computacionales en mecanismos que simulan la evolución biológica [1].

Los algoritmos genéticos son uno de los tipos de algoritmos evolutivos más utilizados, no siendo así la programación genética, que cuenta con menos resultados probados que estos, pero puede encontrarse entre las técnicas más

prometedoras. Por tanto, sería de gran importancia comprobar la factibilidad del uso de la programación genética en tareas de minería de datos, sin embargo, no existe una propuesta concreta para enfrentar, con esta técnica, la tarea de clasificación. Las experiencias y referencias tenidas en el campo de los algoritmos evolutivos en la minería de datos no son numerosas debido entre otras razones a la novedad de su uso. Algunas referencias encontradas son: En ellos se describe la utilización de técnicas de minería de datos en sistemas de aprendizaje, presentando una herramienta visual para el descubrimiento de conocimiento en forma de reglas de predicción en la mejora de sistemas hipermedia adaptativos y educativos basados en Web. En la búsqueda de software que permitieran realizar tareas de minería de datos, fueron encontradas algunas herramientas entre las que se destacan DBMiner, Weka y Keel [2]. Estos sistemas son de dominio público un tanto populares por su entorno gráfico integrado y otra serie de características significativas. Un inconveniente que presentan estos tipos de herramientas es que son complejas de manejar para una persona no experta en minería de datos [3], [4]. Además, se desarrolla un trabajo para el Descubrimiento de Reglas de Predicción mediante una herramienta gráfica denominada EPRules.

MATERIALES Y MÉTODOS

Este trabajo propone un método para construir un clasificador basado en la evolución de reglas, sesgando la búsqueda hacia regiones de hipótesis comprensibles con alta calidad predictiva mediante programación genética. Para dar una visión más específica del algoritmo que se propone, se detalla a continuación todo el proceso que se llevó a cabo, sustentado por algunos códigos de implementación que facilitan su entendimiento.

En la figura 1 se ilustra el método propuesto de forma general.

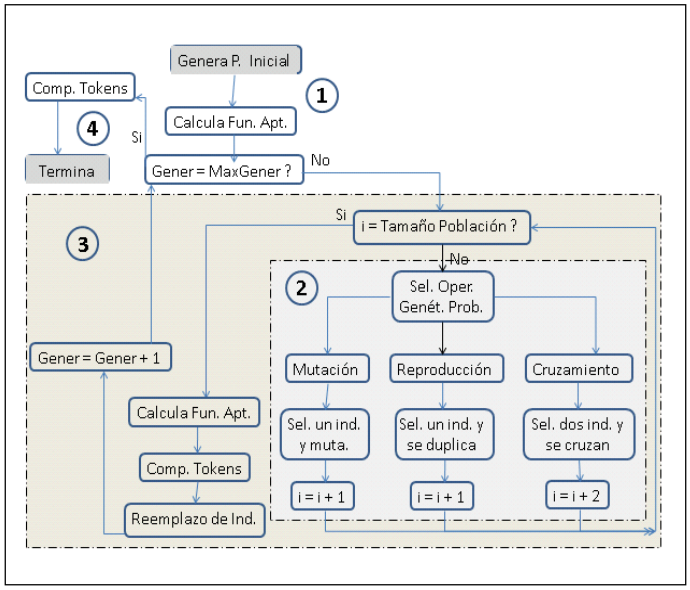

Fig. 1. Esquema que representa el método propuesto

Fase de inicialización

Como se puede apreciar en la figura el algoritmo comienza generando una población inicial de individuos, o sea, reglas de clasificación de forma aleatoria. Una regla, como se ha dicho anteriormente, está formada por un antecedente y un consecuente, y se encuentra representada mediante un árbol binario donde el subárbol izquierdo representa el antecedente y el subárbol derecho el consecuente [5].

Este antecedente de la regla se genera mediante el método de inicialización Grow [6], donde se va formando el árbol, de manera tal, que se vayan creando aleatoriamente nodos internos (no terminales) y nodos hojas (terminales) con los atributos y sus valores mientras se cumpla con la profundidad máxima establecida, todo esto garantizando que dicho árbol tenga una estructura válida para el antecedente de una regla.

Luego se realiza un proceso de búsqueda sobre el conjunto de datos de entrenamiento, en el cual se localiza la clase mayoritaria en dependencia de la cantidad de datos que logre emparejar dicho antecedente y se le atribuye esta clase al consecuente formándose así la regla.

Este proceso proporciona que en un inicio se generen reglas capaces de cubrir al menos un dato. Seguidamente, esta regla pasa a ser evaluada mediante el cálculo de la función de aptitud, donde se utilizaron dos medidas de calidad. Por un lado está el soporte (*S*) [7], cuya forma de cálculo se muestra en la ecuación 1.

$$S = \frac{A}{T} \tag{1}$$

Como se muestra en (1) el término *A* indica el número de instancias que cumple el antecedente de la regla y *T* es el total de instancias utilizadas para entrenar el modelo. Por otro lado se tiene el porcentaje de aciertos (*P*), reflejándose su método de cálculo en la ecuación 2.

$$P = \frac{AC}{A} \tag{2}$$

En (2) se observa que el término AC es la cantidad de instancias que cumplen el antecedente y el consecuente de la regla. Esta medida da la probabilidad que tiene la regla de clasificar los datos correctamente.

Como se decidió utilizar una función multiobjetivo, es decir, un vector de varias medidas de calidad para lograr un balance entre ellas y hacer que se converja hacia el conjunto que está formado por las mejores soluciones (en términos de todos los objetivos individuales, no de cada uno por separado), se forma este vector con las medidas antes mencionadas y se le calcula su norma, para de esta manera, darle cumplimiento al cálculo de la función de aptitud, según la ecuación 3.

$$FA = \sqrt{s^2 + P^2} \tag{3}$$

Es válido destacar que hay otras variantes para calcular una función multiobjetivo, como es el caso de la optimización de Pareto [8]. Por último se adiciona la regla a la población y se repite este proceso hasta que se cumpla con el tamaño de la población predefinido antes de la corrida del algoritmo. De esta forma queda conformada la población inicial.

Fase de evolución de los individuos

Después de culminada la fase uno descrita anteriormente, se pasa a evolucionar en cada generación los individuos, para de esta forma converger a una población con los mejores individuos de acuerdo con el problema que se está analizando, como una característica típica de los *algoritmos evolutivos*. Para desarrollar este proceso se comienza iterando tantas veces como cantidad de generaciones se hayan predefinido en los parámetros de la corrida del algoritmo [9].

En cada iteración se aplica un operador genético (cruzamiento, mutación o reproducción) seleccionado aleatoriamente, en dependencia de las probabilidades que posean. Para la selección de los padres que originarán nuevos descendientes que serán incluidos en la nueva población, se utiliza el método de la ruleta. La idea que se propone en el método es dar a cada individuo una probabilidad de ser seleccionado acorde con su función de aptitud y proporcional a su calidad dentro del espacio muestral de entrenamiento que se esté analizando (cuanto mejor es el valor de la función de aptitud mayor es la probabilidad de ser seleccionado y viceversa) [10]. Después de ser aplicado el operador correspondiente se evalúan estos descendientes (de la misma forma que se describió en la fase uno), se adicionan a una nueva población y así se procede iterativamente hasta que esta última logre alcanzar el tamaño de la población predefinida.

Fase de evolución de las poblaciones

En el proceso iterativo desde la primera generación hasta alcanzar la última prefijada, se realiza una estrategia de corte y remplazo de los peores individuos por otros generados, que garantiza la diversidad de la población, así como su convergencia hacia la mejor solución [11]. Se comienza ordenando la población de individuos por el valor de su función de aptitud, para que se logre garantizar que los mejores individuos sean los primeros en competir por los datos a capturar y con respecto a la nueva población generada que los individuos más aptos sean los escogidos para sustituir a los peores de la actual. Seguidamente se realiza el proceso de *competición de Tokens* a la población actual ya ordenada [12]. En este proceso se analizan cada uno de los individuos para ver a cuántos datos puede capturar y luego los que logran emparejar con dicho individuo son marcados para que otro individuo menos apto que él no pueda apoderarse de este recurso. Posteriormente se le actualiza a cada individuo el valor de la función de aptitud según la ecuación (4).

$$FM = FO \cdot \left(\frac{C}{M} \right) \tag{4}$$

Donde *FM* es la función de adaptación modificada; FO es el valor de la función de aptitud obtenida por el individuo en el proceso de adaptación; *C* es el número de ejemplos que el individuo ha conseguido capturar y *M* es el número máximo de ejemplos que el individuo puede capturar. Finalmente todos los individuos cuyo valor de aptitud sea cero, son eliminados de la población ya que esto indica que no pudieron capturar ningún dato. Por tanto, estos *k* individuos son redundantes y remplazados por los *k* primeros individuos de la nueva población generada, lo que puede aportar un mayor grado de diversidad a la población y además proporcionar cambios adicionales para la generación de buenos individuos.

Fase de culminación

Como conclusión se vuelve a aplicar el proceso de *competición de Tokens* a la última población que se obtiene, para de esta forma culminar el algoritmo con un conjunto de reglas libres de redundancias, y con alto nivel de precisión y que puede tener un número menor de reglas que el tamaño de la población predefinida [13].

RESULTADOS Y DISCUSIÓN

Se ha desarrollado una herramienta específica con el objetivo de facilitar el proceso de descubrimiento de reglas de clasificación y darle una aplicación al algoritmo que se propone en este trabajo. La herramienta implementada tiene una entrada de datos a procesar mediante ficheros con un formato específico (.arff). La herramienta presenta además una amigable e intuitiva interfaz visual, lo que permite que el usuario cuente con diferentes facilidades al trabajar con la misma. A continuación se detallan estas características a las que se hace alusión.

Primeramente se puede decir que el modelo generado, o sea, el conjunto de reglas de clasificación que se obtiene, se muestra de forma clara y precisa, acompañado de diferentes valores estadísticos que se calculan asociados al proceso de validación del modelo obtenido, como son: la cantidad de instancias correctamente clasificadas, las incorrectamente clasificadas, así como las no clasificadas. Además, se muestra la matriz de confusión, que es otra forma de evaluación de un modelo de clasificación [14].

Por otra parte, la herramienta permite realizar varias corridas, es decir, almacena los diferentes modelos que pueden ser generados, para que el usuario sepa en todo momento los resultados de las corridas realizadas y la posibilidad de salvarlos en ficheros.

Las reglas de clasificación están compuestas por un antecedente y un consecuente, donde este último posee una sola condición. Una vez que se ha generado un modelo, se tiene la posibilidad de, dada las características de una

nueva instancia, decir a qué clase pertenece, o sea, clasificarlo de acuerdo con el modelo generado. Para esto primeramente se muestran, de manera dinámica, los atributos con sus respectivos valores para que estos sean seleccionados de acuerdo con las características que posea la instancia a clasificar, todo esto evita que se haga un proceso de validación de entrada de datos. Una vez obtenidos los valores de los atributos de la nueva instancia que se va a clasificar, se compara cada regla del modelo con estos nuevos valores de la instancia, para ver cuál empareja o está contenida y finalmente si se encuentran una o varias reglas que cumplan la condición anterior y presenten un único valor de clase, este se le asigna a la nueva instancia, de lo contrario se emite que no se puede clasificar dicha instancia.

Análisis comparativo de resultados obtenidos

Se han realizado diferentes pruebas orientadas a comparar y validar los resultados del algoritmo que se propone en la tarea de descubrimiento de conocimiento, *clasificación*. Para esto se determinó la comprobación de los resultados obtenidos, realizando dos tipos de corridas diferentes en relación con la cantidad de datos usados para entrenar y probar los modelos generados por cada uno de los ficheros de muestra seleccionados de las bases de datos (Contact-Lenses, Wisconsin, Tic-Tac-Toe, Car) del UC Irvine Machine Learning Repository [15], cuyas características principales quedan resumidas en la tabla 1. También se realizó un análisis comparativo entre el algoritmo propuesto y otros implementados en una de las herramientas estudiadas, que generan modelos formados por reglas de producción y basan su funcionamiento en enfoques evolutivos.

Para estos análisis se decidió tomar un tamaño de muestra de población inicial de 100, con un número de generaciones de 200 y las probabilidades de los operadores genéticos *reproducción*, *cruzamiento* y *mutación* fueron de 0,1, 0,7 y 0,1 respectivamente, quedando reflejados estos parámetros en la figura. 2

Los tipos de corridas utilizadas son los siguientes:

1. El mismo conjunto de datos de entrenamiento y prueba.
2. El conjunto de datos se particiona en dos, un 50 % para el entrenamiento y el resto para prueba.

Tabla 1
Características de los ficheros utilizados

Nombre del fichero	No. de atributos	No. de clases	No. de ejemplos
Contact-Lenses	5	3	24
Wisconsin	10	2	683
Tic-Tac-Toe	10	2	958
Car	7	4	1728

Al realizar las diferentes corridas para cada fichero analizado se obtuvieron los resultados que se expresan en la tabla 2.

Como se puede observar en la tabla 2, los porcentajes correctamente clasificados, analizados para cada fichero, se comportan de forma similar en los dos tipos de corridas que se efectuaron, solo disminuye en un rango de 0,5 % a 5,3 %, lo que demuestra que la disposición de los datos de entrenamiento y prueba de forma variable, no afecta en grandes porcentajes la precisión de clasificación.

Por otra parte se determinó realizar una comparación con dos algoritmos de clasificación que generan modelos basados en reglas de producción de la herramienta KEEL (LogenPro y PGIRLA).

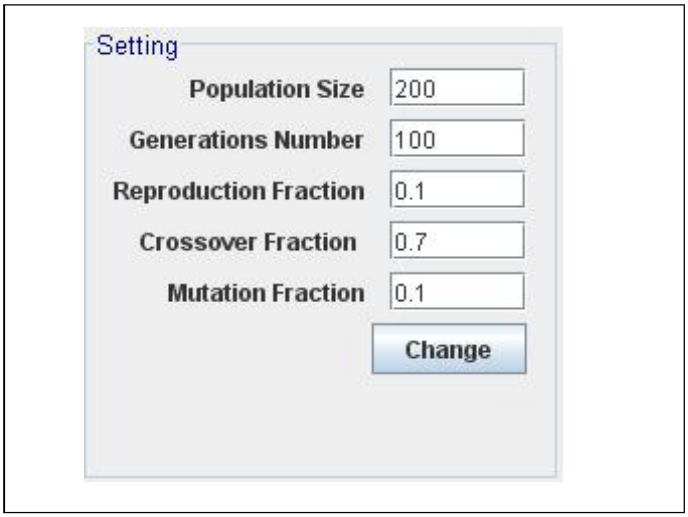

Fig. 2. Parámetros para la corrida del algoritmo

Tabla 2
Características de los ficheros seleccionados

Nombre del fichero	Tipo de corridas	Total de reglas	Ejemplos correctamente clasificados (%)	Tiempo (S)
Contact-Lenses	1	17	95,8	0,71
	2	10	33,3	0,46
Wisconsin	1	91	78,6	7,24
	2	56	78,1	4,09
Tic-Tac-Toe	1	200	64,9	11,39
	2	186	61,8	6,5
Car	1	55	70,0	16,8
	2	74	64,7	10,4

Revista Cubana de Ingeniería. Vol. IV, No. 2, mayo - agosto, 2013, pp. 37 - 42, ISSN 2223 -1781

El primero de estos algoritmos está implementado mediante programación genética y el segundo está basado en algoritmos genéticos, por lo que ambos presentan características evolutivas propias muy parecidas al análizado en este trabajo, lo que facilita una mejor comparación. Al realizar estas pruebas se obtuvieron los resultados que se muestran en la tabla 3.

Como se puede apreciar en las dos tablas anteriores, los resultados obtenidos con el algoritmo que se propone en cuanto a la medida de precisión de clasificación que se presenta, comparándolos con los que arrojan LogenPro y PGIRLA son muy similares y en tres casos arroja los mejores resultados, lo que da una medida de validez del algoritmo que se propone.

Tabla 3
Resultados de las corridas con los algoritmos y ficheros seleccionados

Algoritmos	Contact-Lenses	Wisconsin	Tic-Tac-Toe	Car
Algoritmo propuesto	95,8	78,6	64,9	70
LogenPro	83,3	62,8	69,2	69,2
PGIRLA	58,3	78,4	56,2	56,2

CONCLUSIONES

Para garantizar la extracción de conocimiento como parte del proceso de minería de datos en análisis biológicos, se requiere de novedosos algoritmos que implementen alta eficiencia en sus resultados.

Con la introducción de *algoritmos evolutivos* enfocados a la *programación genética* es posible potenciar resoluciones para la minería de datos que contribuyan a la extracción de conocimientos.

Con la introducción en la práctica social del algoritmo para la generación de reglas de clasificación basadas en programación genética, se pudo realizar un análisis comparativo de los resultados obtenidos con otros algoritmos existentes demostrándose la efectividad de la propuesta.

REFERENCIAS

1. **GUILLERMO, M.; MEDA, M.** "Integración de minería de datos y sistemas multiagente: un campo de investigación y desarrollo". *Ciencias de la Información*. 2010, vol. 41, núm. 3, septiembre - diciembre, pp. 53 - 56. ISN: 1606 - 4925.
2. **CARLOS, CASTRO; GARCÍA ENRIQUE.** *et al*. "Herramienta autor indesahc para la creación de cursos hipermedia adaptativos". *RELATEC*. 2004, vol. 3, núm. 1, ISSN 1695-288X.
3. **SONIA, O.** "Aprendizaje de reglas encadenas para la creación de grafos conceptuales", *Polibits*, ene./ jun. 2010, núm.4. ISSN 1870-9044.
4. **ISEL, G. G.; GONZALO N. R.,** *et al*. "Aplicación de sistemas neuroborrosos a problemas de resistencia anti-viral del VIH". *RCCI*, abril-junio 2012, vol. 6, núm. 2, ISSN: 1994-1536.
5. **RAÚL, C. P.** Un algoritmo genético especializado en planeamiento de redes de distribución, Parte I. Fundamentos técnicos del algoritmo. *Ingeniería Energética*, enero - marzo 2010, vol. XXXII, núm. 1, pp. 72 - 76. ISSN 1815 - 5901.
6. **ÁNGEL, G. B.** *Computación Evolutiva (CE), Programación Genética, Evolución Gramatical, Programación por Expresión Genética*. Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, 2008, 20-45. pp.
7. **CÉSAR, H.; MARTÍNEZ, C.** *et al*. *Selección de medidas de evaluación de reglas obtenidas mediante programación genética basada en gramática*. Universidad de Córdoba. , [En línea]. 2008. [Consultado el 23 enero 2013]. Disponible en: www.lsi.us.es/redmidas/Capitulos/LMD23.pdf
8. **SALTO MOLINA, A.** Analysis of Distributed Genetic Algorithms for Solving Cutting Problems. (ITOR), 2006, vol. 13, núm. 5, pp. 403-423. ISSN 0969-6016.
9. **JOSÉ, M.; DELGADO**, **R.** *et al*. "Algoritmo genético aplicado a la programación en talleres de maquinado" *Ingeniería Mecánica*, septiembre-diciembre 2012, vol. 15, núm. 3, p p. 201-212. ISSN 1815-5944.
10. **ROBERTO, P. C.; JONATAN, G. P.** *et al*. Un algoritmo genético paralelo que combina los modelos de grillas e islas para encontrar soluciones óptimas cercanas al problema del agente viajero, *Revista en avaces en Sistemas e Informática*, diciembre 2008, vol. 5, núm. 3, pp. 13-19. ISSN: 1909-0056.
11. **SERRANO, G.** Herramienta para la generación de reglas de asociación basada en un algoritmo de Programación Genética. UCI: Ciudad de la Habana, 2007. 24-60. p.
12. **ILBER, A. R.** "Design of a Adaptive Controller Based in Genetic Algoritms", *Revista Colombiana de Tecnologías de Avanzada*. 2008, vol. 2 núm. 6, ISSN: 1692-7257.
13. **JOSÉ, D. M.; LILIANA, K. P.** "Un algoritmo genético hibrido y un enfriamiento simulado para solucionar el problema de programa de pedidos Job Shop", *Revista EIA*, julio 2010, núm. 13, p. 39-51, ISSN 1794-1237
14. **ROMERO, C.; SEBASTIÁN**, **V.** *et al*. "Aplicación de algoritmos evolutivos como técnica de minería de datos para la mejora de cursos hipermedia adaptativos basados en Web RIED", diciembre 2008, vol. 6, núm. 2. ISSN: 1138-2783.
15. UNIVERSITY OF CALIFORNIA-IRVINE. *Machine Learning Repository (Repositorio de bases de Datos UCI*), [En línea]. 2012. [Consultado el 23 enero 2013]. Disponible en: http://archive.ics.uci.edu/ml/

AUTORES

José Leandro González
Ingeniero Informático, Universidad de las Ciencias Informática (UCI), La Habana, Cuba

Omar Mar Cornelio
Ingeniero Informático, Departamento de Programación, UCI, La Habana, Cuba

Proposed Genetic Classification Algorithm

Abstract

The extraction of knowledge is the basis for decision making in various processes, data mining being an area of fundamental knowledge to attain that end. Among the processes used today are Artificial Intelligence and Statistical Analysis techniques implemented by Association, Clustering, classification that are developed using stochastic methods, heuristics, neural networks. However analysis to biological problems give temporary solutions representing low degree of generalization. This research describes a solution to the problem created incident on the classification technique based on the creation of a genetic programming algorithm for generating rules proposing four stages in its implementation which passes from initiation, through the evolution of as individuals, stock assessment to completion. It also validates the efficiency of the algorithm making a comparative analysis of the solutions Pgirla LogenPro and demonstrating the efficiency of the proposal.

Key words: algorithm, classification, genetic programming, rules